SIM Trust Parameters

Mobile Service Technologies

Jane Dashevsky

Senior Software Engineer

Intel Corporate Technology Group

Edward C. Epp

Senior Software Engineer

Intel Corporate Technology Group

Jose Puthenkulam

Senior Software Engineer

Intel Corporate Technology Group

Mrudula Yelamanchi

Software Engineer

Intel Corporate Technology Group

(Rev 1.5, January 2003)

Intel
Research &
Development

Table of Contents

Acknowledgements

The authors would like to thank Carl Ellison for his free spirited and insightful comments and suggestions, and Gary Graunke for his detailed lessons on TRS methods and related security issues. We would also like to acknowledge the significant efforts made by Roger Chandler for editing the paper and considerably improving it in the process. We would also like to thank everyone who reviewed the paper and contributed to our efforts overall.

Acronyms

CAPI	Crypto API
CDMA	Code Division Multiple Access
ESA	Enhanced Subscriber Authentication
ESP	Enhanced Subscriber Privacy
ETSI	European Telecommunications Standards Institute
3GPP	3rd Generation Partnership Project
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HLR	Home Location Register
IMEI	International Mobile Equipment Identifier
IMSI	International Mobile Subscriber Identity
ITU	International Telecommunication Union
ME	Mobile Equipment
MS	Mobile Station
PDA	Personal Digital Assistant
SGSN	Serving GPRS Support Node
SIM	Software Identity Module
SRES	Signed RESponse
TMSI	Temporary Mobile Subscriber Identity
TPM	Trusted Platform Module
TRS	Tamper Resistant Software
TSS	TCPA Software Stack
UICC	Universal IC Card
UMTS	Universal Mobile Telecommunications System
USIM	UMTS SIM
VLR	Visiting Location Register
WLAN	Wireless Local Access Network
WWAN	Wireless Wide Area Network

1 Introduction

The mobile PC platform is becoming a popular means of accessing cellular data networks using 2.5G and 3G wireless technologies. Smart card based SIM (Subscriber Identity Module) technology and its derivatives are the established means for securely accessing cellular networks from mobile computing devices. However, this type of SIM implementation, is not the only alternative for these platforms.

Running software SIM on a mobile PC would allow significant expansion of service by introducing new use scenarios, seamless connectivity across a variety of networks, and improved the end-user experience.

The objective of this white paper is to evaluate the different alternative software SIM implementations for the PC platform and discuss their strengths and weaknesses.

1.1 Scope of this document

Ultimately hardware will play a critical part of any reasonable security solution., so specialized security hardware support will receive some attention, but this paper will primarily evaluate currently available software-based security solutions

The traditional mobile handset is a reliable benchmark for evaluating client side trust in cellular networks. We will draw parallels with it and the PC platform. Figure 1 captures the major components in the traditional closed handset. Operators control its provisioning and users can only install a few components like ring tones, and graphic files. Even with modern handsets, which come with Java* capabilities, operators restrict customization to certified applications.

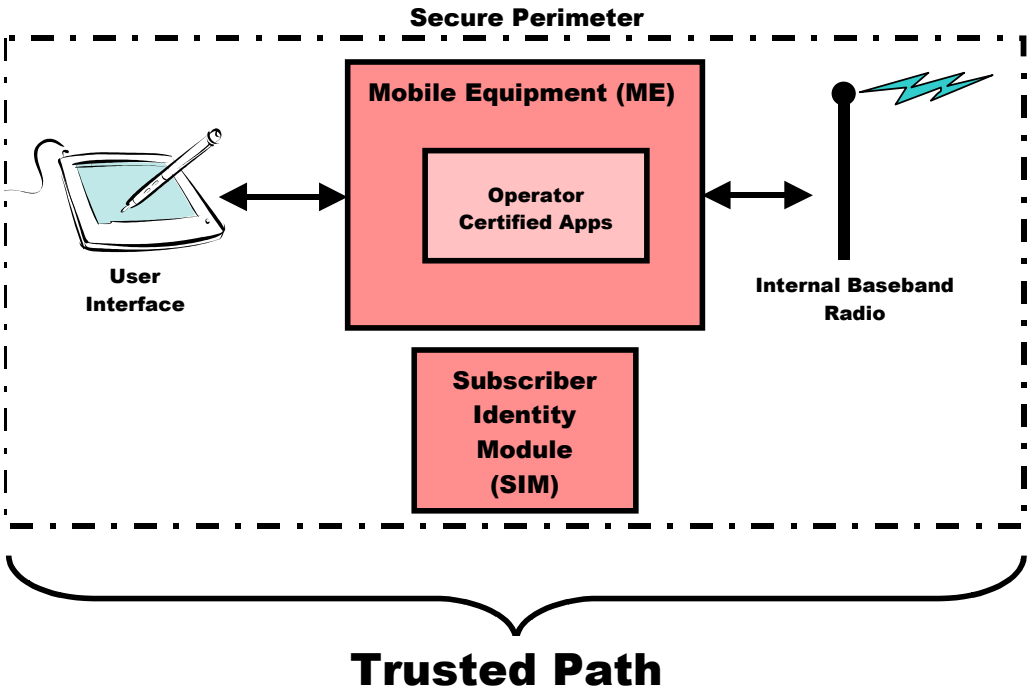


Figure 1 Architecture for closed handset

1.2 Goal of this document

Our goal in this document is to determine the trust parameters and trust/cost trade-offs for software implementation of SIM using various software and hardware security technologies on traditional PCs, as compared to mobile handsets. We will also explore the issues involved with software SIM solutions on PC platforms. In particular, current PC hardware and software cannot support a trusted path from subscriber or operator to the SIM. Simply adding a SIM smart card to a PC is not enough because the "trusted path" is missing. Figure 2 illustrates how the secure perimeter has shrunk to enclose only the SIM module.

There are multiple points where operating systems and applications can be compromised: viruses, Trojan Horses, Web downloads, and software installs or upgrades. In an attempt to improve security some vendors have placed a GPRS transmitter and SIM in one PCMCIA card. Unfortunately this strategy only moves the radio interface within the security perimeter. The path from the user interface to card is still open.

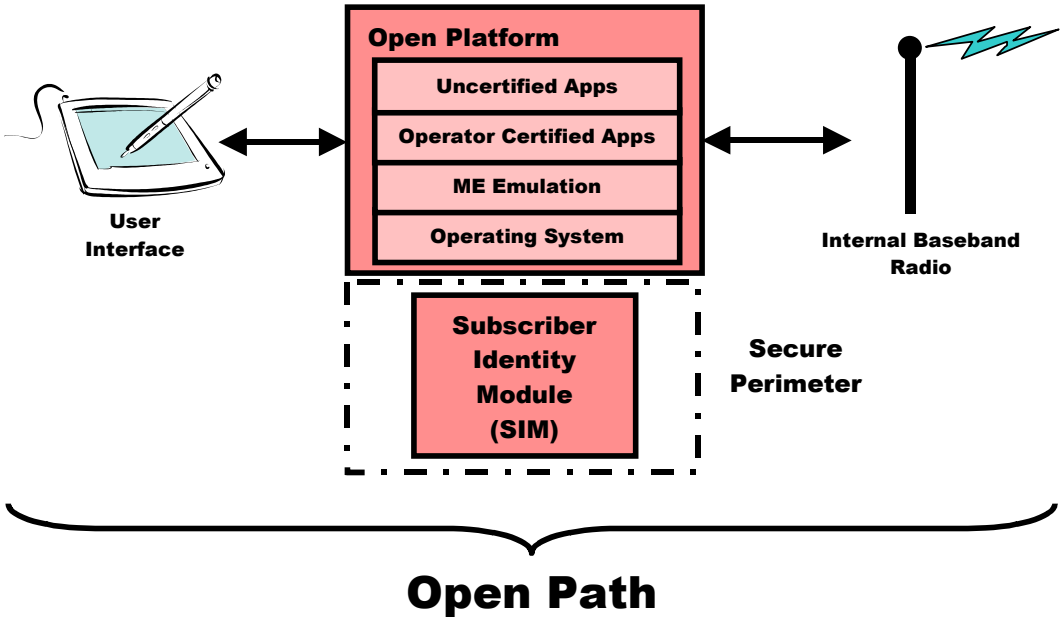


Figure 2 Architecture for open system (desktop, laptop, handheld)

1.3 Brief Outline

This paper will give the reader a background in security problems and threats in WWAN networks (GSM, GPRS, and 3G). It will cover security mechanisms used by network operators and will analyze the level of trust provided by them. This analysis will then be used to derive trust requirements and capabilities of interest to operators and subscribers.

It will then describe security technologies that provide these capabilities and will evaluate three categories of operating system architectures. These architectures were chosen because they provide a significant basis for evaluating the open PC platform trust models and can be created from off-the-shelf components. They include:

- Those without isolated environments
- Environments that isolate users
- Virtual machine solutions that support isolated environments

Some of the security technologies, when combined with these operating systems, form solutions that will also be considered, compared, and summarized.

2 Security in GSM, GPRS, 3G Cellular Networks

This section looks at the security in cellular networks with emphasis on the Mobile Station /Client.

2.1 GSM Security Model

The use of radio as a voice and data transmission method introduces a number of potential issues in delivering fully secure communications. In order to be successful, GSM networks need to establish a level of security comparable to the public switched telephone network. Operators must be able to issue bills to the right people, and must be able to issue services that will not be compromised. This requires a strong authentication method. The customer requires confidentiality and anonymity for their transmissions. Also, operators should be unable to compromise each other's security, whether inadvertently or because of competitive pressures.

The design of a GSM system takes into account the environment and employ secure procedures such as:

- The generation and distribution of keys,
- Exchange of information between operators,
- The confidentiality of the algorithms.

2.1.1 GSM security Basics

The security services provided by GSM are[i] [ii]:

- ***Anonymity*** -- it is not easy to identify the user of the system.
- ***Authentication*** -- the operator knows who is using the system for billing purposes.
- ***Signaling and User Data Protection*** -- sensitive information on the signaling channel, such as telephone numbers, is protected over the radio path and to ensure privacy of the user data over the radio path.
- ***Authorization*** -- access to certain functions or information is established.

2.1.1.1 Anonymity

GSM networks use Temporary Mobile Subscriber Identities (TMSI) to ensure that the identity of subscribers remains protected on the cellular network. There is a short window of opportunity to determine the identity of a subscriber when the mobile station makes initial contact with the network. When a mobile station makes contact with the network, it must provide its International Mobile Subscriber Identity (IMSI). The IMSI contains the personal subscriber number, the name of its home network and code of the country in which its subscription is based. Once the network is finished using this information to identify the subscriber, the mobile station is assigned a TMSI. After this point, anonymity is maintained.

2.1.1.2 Authentication

Authentication is used to identify the user (or holder of a Smart Card) to the network operator. It uses a technique that can be described as a "Challenge and Response". It is based on a shared secret between subscriber's home network Home Location Register (HLR) and the subscribers SIM.

A random challenge is issued to the mobile device; the mobile device encrypts the challenge using the authentication algorithm (A3) and the key assigned to the device, and sends a response back. The operator can check that, given the key of the mobile device, the response to the challenge is correct.

Eavesdropping on the radio channel reveals no useful information, as the next time a new random challenge will be used. Authentication can be provided using this process. A random number is generated by the network and sent to the mobile. The mobile device uses the random number R as the input (Plaintext) to the encryption, and, using a secret key unique to the mobile device (K_i) transforms this into a response Signed RESponse (SRES) (Ciphertext) which is sent back to the network.

The network can check that the mobile device really has the secret key by performing the same SRES process and comparing the responses with what it receives from the mobile device.

2.1.1.3 User Data and Signaling Protection

The response is then passed through an algorithm A8 by both the mobile and the network to derive the key Kc used for encrypting the signaling and messages to provide privacy (A5 series algorithms).[iii]

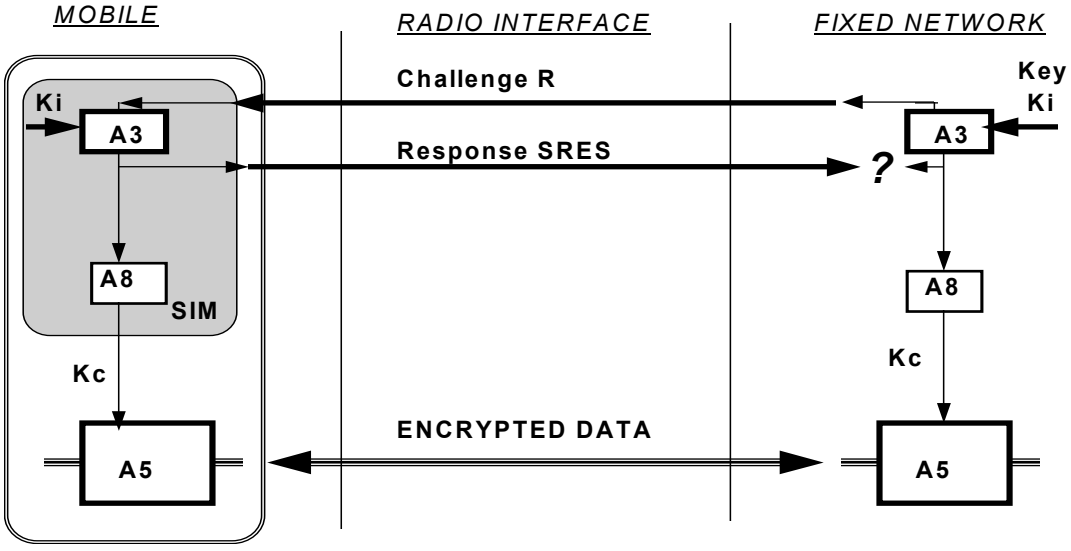


Figure 3 Encryption for GSM

2.1.2 Security support for Roaming

The authentication algorithm A3 is an operator option, and is implemented within the smart card (known as the Subscriber Identity Module or SIM). So that the operators may

inter-work without revealing the authentication algorithms and mobile keys (K_i) to each other, GSM allows triplets of challenges (R), responses (SRES) and communication keys (Kc) to be sent between operators over the connecting networks.

The A5 series algorithms are contained within the mobile equipment, as they have to be sufficiently fast and are therefore hardware. There are two defined algorithms used in GSM known as A5/1 and A5/2. The enhanced Phase 1 specifications developed by ETSI allows for inter-working between mobile devices containing A5/1, A5/2 and unencrypted networks. These algorithms can all be built using a few thousand transistors, and usually takes a small area of a chip within the mobile.

2.1.3 Other GSM security mechanisms

2.1.3.1 SIM Security

The smart card, an intelligent token, is a credit card–sized plastic card with an embedded circuit chip. It provides not only memory capacity, but computational capability as well. The self-containment of the smart card makes it more resistant to attack, as it does not need to depend on potentially vulnerable external resources. The smart card itself is easy to use, portable, unique and designed not to be replicated.

One of the fundamental problems in securing computer systems is the need for tamper-resistant storage of keys. Smart cards provide this functionality, as well as the ability to upgrade and/or replace a security solution when it becomes compromised. Smart cards give you mobility in a secure way. They can also act as an identification card, which is used to prove the identity of the cardholder.

A smart card provides secure execution along with secure storage. With a smart card, the private key never leaves the card and is completely inaccessible from outside the card. All cryptographic functions requiring use of the private key for secured Internet browsing and secured e-mail delivery – digital signatures and decryption of the session keys – take place on the smart card by the onboard microprocessor, and only the results are passed back to the host PC.

Access control is provided on a file-by-file basis. Adding accessing conditions and file status fields in the file header enhances the attribute of each file. Moreover, file lock is also provided to prevent the file from being accessed. These security mechanisms and algorithms ensure a logical protection provided by the smart card.

2.1.3.2 Theft Prevention (IMEI)

In a GSM network, the customer subscription and authentication capability is contained within a smart card (SIM, Subscriber Identity Module). Any mobile device will take on the identity of a subscriber by the insertion of a smart card. The mobile devices now become attractive items to steal, as they can be used with another SIM card.

To prevent this, GSM has specified an International Mobile Equipment Identifier (IMEI). To an operator, at first evaluation, it may seem as though the stolen mobile devices have no effect, as they do not affect a subscription. However, there will be problems involving an increase in customer support, lack of motivation to store sensitive data on mobile

devices, and the possibility that GSM handsets will become expensive to insure in mass quantities.

An Equipment Identity Register (EIR) exists in each network, with Black, White and Grey Lists for stolen or non-type approved mobile devices, valid mobile devices and mobile devices that need tracking respectively. Grey lists are for local tracking of mobile devices within a network.

GSM has defined a procedure so that approved, lost or stolen mobile IMEIs can be communicated to all other operators. A Central Equipment Identity Register has been (CEIR) proposed. Type approval authorities issue white list numbers (random ranges of valid IMEIs) to mobile manufacturers, and manufacturers inform the CEIR when the mobile devices are released to market. All operators are able to post their black lists to the CEIR, and in return collect a consolidated list of all operators black and white lists.

By this method stolen or invalid mobile devices can be quickly barred throughout the world.

2.1.4 Problems with GSM security

Active attacks

A false element can masquerade as a base station to terminals and as a terminal towards a network. The main objectives of the attacker are to eavesdrop, steal a user's connection, and access/manipulate data.

Weak Encryption Algorithms[iv]

Key lengths used in GSM are too short. The key size is reduced to 54 with the last ten bits as 0's. Encryption algorithm COMP 128 has been broken. Replacement of encryption algorithms is quite difficult. There is also a lack of confidence in the algorithms as they are closed.

Key Transmission and Encryption

Cipher keys and authentication values are transmitted within and between networks (IMSI, RAND, SRES, Kc) making it susceptible to. There is no end-to-end encryption. Encryption is terminated too soon at the edge of network to BTS. This is designed to be only as secure as the fixed network.

Unilateral authentication

Only user authentication to the network is provided. There is no means to identify the network to the user.

Unsecured terminal

IMEI is an unsecured identity

Lawful Interception & Fraud

This was considered only as afterthought.

Channel Hijack

Protection against radio channel hijack relies on encryption. However, encryption is not used in some networks.[v]

2.2 GPRS Security

The main function of a GSM/GPRS network is to support and facilitate the transmission of information, whether it is voice or data. The type of information that must be protected on a GSM/GPRS network includes the following:

User Data – This is either voice or non-voice data sent or received by users registered on a GSM/GPRS network.

Charging Information – Information collected from the SGSN and GGSN used to bill for non-voice services.

Subscriber Information – This information is stored in the mobile station, the HLR and the VLR. This is customer specific information for subscribers and roaming users.

Technical Information of the GSM/GPRS Network – This information describes and lays out the GSM/GPRS network architecture and configuration.

Mobile service providers or operators are ultimately responsible for implementing and enforcing security across their GSM/GPRS networks. Some of the hardware on a

GSM/GPRS network comes packaged with security features such as data encryption and user authentication techniques.

2.2.1 Standard GPRS Security Services

The standard security services provided by GPRS include the following:

2.2.1.1 Anonymity

GPRS networks use the same mechanism as the GSM networks. The details for this can be found in 2.1.1.2.

2.2.1.2 Authentication

The basic authentication mechanism used in GPRS networks is the same as GSM. On the mobile terminal side the mechanism is the same and on the network the difference is that authentication is provided by SGSN. During this interaction between the mobile station and the SGSN, the subscriber's private key is never transmitted over the radio interface to the SGSN for use, thus ensuring that the private key remains private.

2.2.1.3 Signaling Protection/User Data Protection

Signaling and user data transmitted over the GPRS IP-backbone and over the radio path is protected from interception and eavesdropping through encryption methods. From the mobile terminal perspective this is the same as in GSM. On the network the data is encrypted between mobile terminal and the SGSN (and not the base station). Data is encrypted using an algorithm called GPRS-A5, a modified version of the A5 algorithm used to encrypt voice communications over GSM networks.

2.2.2 Security issues in GPRS

GPRS has inherited most of the security threats that exist in the GSM system. In addition GPRS encounters new and bigger challenges, since it employs IP technology and it is connected to the Internet. Insecure public Internet, attacks on the data at the air interface, or at the operator-level with the handling of data that is transmitted or stored in the

network, can be a threat. From the mobile terminal perspective the security issues in GPRS are the same as GSM.

2.3 3d Generation Security: 3GPP and 3GPP2

3GPP (3rd Generation Partnership Project) and 3GPP2 (The Third Generation Partnership Project 2) are collaborative third generation (3G) telecommunications standards-setting projects. 3GPP covers 3rd Generation Mobile Systems based on evolved GSM core networks and the radio access technologies that they support, as well as maintenance and development of existing GSM, GPRS, and EDGE. 3GPP2 was born of ITU's International Mobile Telecommunications "IMT-2000" initiative, covering high speed, broadband, and Internet Protocol (IP)-based mobile systems comprising North American and Asian interests developing global specifications for ANSI/TIA/EIA-41[vi] [vii]

2.3.1 3G Security

3G cellular technologies are improving security, based on learnings from 2G (GSM). In general, 3G standards are much more mature in understanding and defining security requirements and the classification of threats. [viii]

As specified in 3GPP TS 33.120[ix], there are three key principles behind 3G security:

- It is built on GSM security;
- It improves on the security of 2nd generation systems (corrects real and perceived weaknesses of GSM);
- It offers new security features and secures new services offered by 3G.

2.3.1.1 3G security compared to GSM security

The following important changes were made in 3GPP security, as compared to GSM [3GPP]:

- A change was made to defeat the false base station attack. The security mechanisms include a sequence number that ensures that the mobile device can identify the network.
- Key lengths were increased to allow for the possibility of stronger algorithms for encryption and integrity.
- Mechanisms were included to support security within and between networks.
- Security is based within the switch rather than the base station as in GSM. Therefore links are protected between the base station and switch.
- Integrity mechanisms for the terminal identity (IMEI) have been designed in from the start,
- Mutual authentication
- When roaming between networks, such as between a GSM and 3GPP, only the level of protection supported by the smart card will apply. Therefore a GSM smart

card will not be protected against the false base station attack when in a 3GPP network.

- USIM replaced SIM.

The following items from the above list are particularly relevant for implementing SIM on the open platform: stronger algorithms, longer keys, IMEI, authentication vectors.

2.3.1.2 Communications node security

There is a special provision in 3G for Requirements on the terminal and USIM[x]. Here is a brief outline:

- USIM: access control to USIM data; some data / algorithms can be only used within USIM.
- Terminal: deter the theft of terminals; bar a particular terminal from accessing 3G services; difficult to change the identity of a terminal to circumvent barring it from accessing 3G services.

2.3.1.3 User Module and SIM Algorithms

3GPP algorithm requirements are covered in TS 33.105[xi]. There is a very strong requirement covering resilience: the functions should be designed for continued use over a period of at least 20 years. Successful attacks with a work load significantly less than an exhaustive key search, through the effective key space, should be impossible. This spec further details the security procedures supporting authentication and key generation in 3GPP.

2.3.1.4 Authentication of user and terminal

Authentication in 3G is two way and establishes a cipher and an integrity key. The method was chosen in order to achieve maximum compatibility with the current GSM security architecture. It is composed of a challenge/response protocol identical to the GSM subscriber authentication and key establishment protocol combined with a sequence number-based one-pass protocol for network authentication derived from ISO/IEC 9798-4 (section 5.1.1). Key freshness and an authenticated management protocol are provided. All keys are 128 bits, and MACs are 64 bits (except for signaling messages). Air interface encryption applies to all user traffic and signaling. A stream cipher called Kasumi is the default, but null encryption and other algorithms can be used. The ciphering is at a low layer and applies also to the microwave link. The integrity mechanism is similar in scope and strength, but integrity is mandatory. However, signaling MACs are only 32 bits long. The mobile initiates algorithm negotiation. Re-authentication is performed when entering a new network or when new keying material is needed.

CDMA2000 networks use a two-fold authentication mechanism and combine an HLR-based method with the Radius (password-based) authentication for data.

2.3.1.5 Other algorithms

The following algorithms are specified by the 3GPP standard for running in ME:

f9 (Integrity algorithm) for authenticating the data integrity and data origin of signaling data;

f8 – stream cipher for encryption of data frames.

2.3.2 Analysis of attacks / threats in 3G

3GPP specification TS 33.900 has detailed description of potential attacks and how 3G counteract them. Capabilities required for performing attacks are: eavesdropping, impersonation of a user, impersonation of the network, man-in-the-middle, compromising authentication vectors.

2.3.3 Classification of Security Threats in 3G

3GPP TS 21.133 specification "Security Threats and Requirements" uses the following categories of threats.

Unauthorized access to sensitive data (violation of confidentiality): Eavesdropping, Masquerading, Traffic analysis, browsing for sensitive information, Leakage, Inference (observing reaction from a system).

Unauthorized manipulation of sensitive data (Violation of integrity): Manipulation of messages, Disturbing or misusing network services (leading to denial of service or reduced availability), Intervention, Resource exhaustion (overloading the service), Misuse of privileges, Abuse of services, Repudiation.

Unauthorized access to services: masquerading or misusing access rights.

The specification considers three possible points of attack: Radio interface, other parts of the system, Terminals and UICC/USIM. The latter is of special interest to us.

The following threats are associated with attacks on the terminal and UICC/USIM:

- Use of a stolen terminal and UICC
- Use of a borrowed terminal and UICC
- Use of a stolen terminal
- Manipulation of the identity of the terminal
- Integrity of data on a terminal (Access to the terminal may be obtained either locally or remotely, and may involve breaching physical or logical controls)
- Integrity of data on USIM
- Eavesdropping the UICC-terminal interface
- Masquerading as an intended recipient of data on the UICC-terminal interface
- Manipulation of data on the UICC-terminal interface
- Confidentiality of certain user data in the terminal or in the UICC/USIM
- Confidentiality of authentication data in the UICC/USIM.

The threats have been analyzed and evaluated regarding the combined likelihood of occurrence and severity of impact. The threat analysis and the assessment of risks have followed the procedure outlined by ETSI, with extensive use of the collected experiences of operators of first and second-generation systems.

The following threats are considered major:

- Eavesdropping user traffic
- Masquerading as a communications participant
- Passive traffic analysis (Intruders may observe the time, rate, length, sources or destinations of messages on the radio interface to obtain access to information)
- Masquerading as a user
- Use of a stolen terminal and UICC
- Use of a stolen terminal
- Manipulation of the identity of the terminal (Users may modify the IMEI of a terminal and use a valid USIM with it to access services)
- Misuse of user privileges (Users may abuse their privileges to gain unauthorized access to services or to simply intensively use their subscriptions without any intent to pay)
- Confidentiality of authentication data in the UICC/USIM.

The following threats are considered to be medium:

- Masquerading as another user
- Eavesdropping signalling or control data
- Manipulation of the terminal or USIM behaviour by masquerading as the originator of applications and/or data
- Masquerading as a serving network (Intruders may impersonate a serving network, or part of a serving network's infrastructure, perhaps with the intention of using an authorised user's access attempts to gain access to services himself)
- Integrity of data on a terminal
- Integrity of data on USIM

Most of the significant threats can be categorized into a small number of groups: Masquerading, Eavesdropping, & Subscription fraud.

Based on the threat analysis, 3GPP formulated the set of security requirements that cover service access, service provision, system integrity, terminal / USIM, and protection of personal data.

2.3.4 Security in 3GPP2

3GPP2 introduced Enhanced Subscriber Authentication (ESA) and Enhanced Subscriber Privacy (ESP) [3GPP2]. It defined requirements for the cdma2000 Air Interface to support ESA and ESP. [xii]

The enhanced security capabilities address:

a. Unauthorized use (i.e. theft) of service, and unauthorized communications to the MS (i.e., unauthorized base stations attempting to control the MS or retrieve any information from the MS)

b. Unauthorized monitoring of subscriber traffic (i.e., unauthorized eavesdropping).

The cryptographic strength of the ESA process is independent of the cryptographic strength of the ESP process. Compromise of the ESP process does not weaken the ESA process.

Enhanced Subscriber Authentication (ESA)

ESA provides methods for determining the authenticity of any request for service made on an air interface, as well as methods for determining the authenticity of the BS. ESA is supported on all wireless channels and in all MS states in which access to services can be requested. On the control channels, ESA provides the ability to authenticate every message transmitted by a MS or the base station that may compromise the subscriber's security. On dedicated channels, ESA authenticates any message that a MS transmits to request new or different network resources, and any message that a BS transmits that may compromise the ESA security. ESA uses challenge-response authentication (not a mandate); wherein the challenge is random and the response is generated by correspondingly keyed cryptographic algorithms within the MS and the network.

The authentication procedure prevents replay attacks by minimizing the likelihood that authentication signatures are reused. ESA verifies that the MS contains data representing a valid subscription (IMSI or MIN). ESA also verifies the authenticity of the base station. The authentication process permits subscriber identity authentication independent of the MS identity (ESN or IMEI).

Enhanced Subscriber Privacy (ESP)

ESP provides encryption across the air interface to protect subscriber traffic, both voice and data, as well as certain signaling messages, from unauthorized disclosure.

Keys for ESP may be based on the root authentication key. Keys used for ESP are cryptographically decoupled from the keys used for authentication. A compromise of privacy keys does not compromise the root authentication key. The privacy key can be modified in the MS under control of the home system. Keys for ESP are changed with each new security association. Privacy keys for each call are established at the time of authentication of an MS service access. Privacy keys for control channel encryption are established at the time of MS system access, after a successful authentication.

ESP provides a mechanism to easily enhance the algorithm (or algorithms), key generation procedures, or both, in the event the security of ESP is compromised.

3 Trust Requirements for Virtual SIM

This chapter serves as a link between the security analysis for current closed implementations of SIM and the exploration of alternatives. It is organized into three parts.

- Our first step in Section 3.1 is to carefully specify the high-level trust requirements of SIM today, based on the data collected in Chapter 2.
- In section 3.2 we identify major SIM stakeholders and capture objects and processes that are critical to each. The goal of this section is to justify access control to each stakeholder's resources. We partition access control mechanisms into three classes: secure storage, secure execution environment, and secure communications in Section 3.3.
- Section 3.3 translates the access control mechanisms into capabilities that are guaranteed if these requirements are satisfied. Our goal is to derive the simplest set of requirements possible that meet stakeholders' needs and use them to validate solutions in Chapter 5.

3.1 High Level Trust Requirements

In Chapter 2 we presented an analysis of security mechanisms and practices that are currently employed by GSM and 3G service providers, as well as the ones that are reflected in standards. The 3G standard bodies created a detailed security Requirement specification that is particularly useful for our analysis.

Out of these numerous requirements we derive those that are relevant for moving SIM into software. All protocols, algorithms, and data structures remain the same — currently in the existing WWANs. Also, there is no change in air interfaces or network infrastructure due to moving SIM to software. The only change is the environment in which the protocols, algorithms, and data objects are stored and executed – we move them from the closed implementation inside the isolated hardware module into the software implementation on the open platform.

There are two major high-level requirements that need to be met:

1. The service must not be a) stolen, b) duplicated, or c) modified in a way that is not supported by service agreement.
2. Personal user data must be protected.

On a more detailed level, the following requirements must also be met:

1. It must not be possible to copy SIM data (algorithms, keys, other) by an unauthorized party.
2. It must not be possible to alter SIM data by an unauthorized party (protects from changing service parameters and DOS).
3. Unauthorized access to services or use of service must be prevented.

4. There must be protection from denial of service attacks.
5. It must be possible to protect confidentiality of user-related data stored by the user in the terminal or USIM (R5b).
6. Access to USIM (R6a) and to data in a USIM (R6b) must be controlled.

Neither of the requirements stated above is achieved in the current implementations of WWANs. There are multiple faults discovered in GSM systems (see Section 2.1.4), and, although 3G networks are doing a much better job, they are also far from perfect. With this in mind, for each of requirements stated above, the level of strength achieved by the open platform solutions should compare to the levels achieved in current GSM systems.

Porting SIM to software is not intended to improve the security of WWANs. There is no point in making the terminal security stronger than the level that the air interface or the network provides – the overall system security is only as strong as its weakest link. However, if a comparable level of security is achieved, running SIM on a laptop or PDA would allow a significant expansion of service by introducing new use scenarios, seamless connectivity across a variety of networks and improved user experience.

3.2 Access Controls – What We Are Asked to Protect

The requirements listed above focus on who is authorized to access and modify data, execute algorithms, and use services. Access control is a natural mechanism to satisfy these requirements. Within this section we look at example data and processes that are important to the current SIM stakeholder and how that stakeholder wants access controlled.

For each stakeholder, entities are partitioned into data (passive) and processes (active) objects. The stakeholders of interest are operators and subscribers. An object of interest is categorized with the stakeholder that creates or initiates the creation of that object. We are only interested in those objects stored on the subscriber's device.

3.2.1 Operator

1. Data & settings
    a. Data, if made known to unauthorized parties, could result in a theft of revenue. Visibility must be restricted to a small set of local processes specified by the operator. Data is never deleted or modified. A prime example is the private key.
        i. Create – operator at init time
        ii. Modify – no one
        iii. Delete – no one
        iv. Read – selected unmodifiable local processes (for example, signing and encryption processes)
    b. Data if made public has little value. However, its modification could result in denial of service or compromise subscriber privacy (high aggravation for the subscriber and loss of some future chargeable revenue by operator).

An example is MSISDN (Mobile Station International Subscriber Dialing Number).

i. Create – operator at init time
ii. Modify – operator
iii. Delete – operator
iv. Read – authorized third parties

c. Data if made public has little value. However, its modification causes inconvenience (aggravation for the subscriber). An example is preferred language and last number dialed.

i. Create – operator at init time
ii. Modify – subscriber
iii. Delete – subscriber
iv. Read – authorized third parties

d. Data if accessed could result in a loss of revenue for services rendered. Reading this data by operator and authorized clients is necessary. However, reading by unauthorized clients may be an invasion of privacy (resulting in possible revenue loss through law suites or customer base). An example is billing information stored on the ME.

i. Create – operator at init time
ii. Modify – operator, authorized third party
iii. Delete – subscriber if release by operator
iv. Read – operator, subscriber

2. Processes (important processes include those for authentication, encryption, signature, and provisioning.)

a. Processes that if modified or executed by unauthorized clients can result in the theft of services.

i. Provisioning – operator at init time
ii. Continuous provisioning – operator
iii. Disable – operator
iv. Process management (execute, access, halt) – operator, subscriber, and other authorized third parties

b. Processes that, if modified or executed by unauthorized clients, can result in the theft of elementary features. Examples include calendar and special purpose communication applications and options.

i. Provisioning – operator at any time
ii. Continuous provisioning – operator

iii. Disable – operator, subscriber

iv. Process management (execute, access, halt) – subscriber and authorized third parties

3.2.2 Subscriber

1. Data & settings
    a. Data which the subscriber considers private and which the access or modification of which may be an annoyance and/or illegal. An example is an address book.
        i. Create – subscriber at any time
        ii. Modify – subscriber
        iii. Delete – subscriber
        iv. Read - subscriber
    b. Data, if made public, could result in theft. Data whose modification could result in a loss of access to resources. Examples include financial, access, and other private keys.
        i. Create – subscriber at any time
        ii. Modify – no one
        iii. Delete – subscriber
        iv. Read - no one (signing, encryption processes)
2. Processes
    a. Processes that, if modified, can make a service unavailable. For example, features, functions, applications
        i. Provision – subscriber at any time
        ii. Continuous provisioning – subscriber
        iii. Disable – subscriber
        iv. Process management (execute, access, halt) – subscriber and authorized third parties

3.3 Access Control – Classes of Things to Protect

We can classify the items requiring protection, listed above, into storage, execution, and communication. This better enables us to define a set of capabilities that have to be implemented to satisfy our initial set of requirements.

1. *Secure Storage*: Secure storage implies controlled access based on an enforced policy. Data and procedures in the store must be protected against unauthorized

access and modification. An example from the functional requirements is K_i – the shared secret between the subscriber's SIM and the network's HLR (section 2.1.1 and 1a in section 3.2.1). A third party can steal an identity by copying K_i or cause the denial of service by modifying it. Another example comes from the 3G world, where the f9 algorithm is used for data integrity (section 2.3.1). Secure storage is also a mechanism to achieve integrity of software and data.

2. *Secure Execution Environment*: A secure execution environment implies a tamper resistant and secure chain of execution, which can prevent and detect interference. Local and remote clients (human, process, program) must be able to determine that they are executing the processes they expect. In addition, the client must be assured that their data and control inputs made it to the appropriate executable and back without interception or modification. An example from the functional requirements is the A3 algorithm and the SRES data it produces (section 2.1.1 and 2a in section 3.2.1). We must be able to prevent a man-in-the-middle attack where a third party captures the SRES data in route and fakes authentication.
3. *Secure Communication*: Secure communication implies communication that is private from and unmodified by third parties. An example from the functional requirements is the K_C that is generated by the A8 algorithm and used by the A5 algorithm (section 2.1.1 and 1a in section 3.2.2) for encryption. A similar example comes from the 3G world where f8 supports encryption.

3.4 Implementation Mechanisms – Capabilities Required to Protect It

In this section we define the set of capabilities that can be used to implement access control of memory, execution, and communication. These capabilities are used in Chapter 5 to evaluate various SIM solutions. These capabilities may have overlapping features but they are essential for the evaluation of any solution. They should not be viewed as architectural components. A capability may require multiple technologies to implement and a single technology may implement multiple capabilities.

3.4.1 Capabilities

The capabilities needed to support secure storage, execution, and communications are[1]:

- Integrity
- Access policy
- Authenticity
- Trusted path
- Trusted boot

[1] There is an underlying assumption that we have at our disposal a general purpose computing environment on which we can add custom methods to fill out our capabilities. For example, we can add a crypto library that provides encryption and signing functionality.

3.4.1.1 Integrity

One feature identified by the TCPA (Trusted Computing Platform Alliance) for secure environments is integrity.[xiii] The integrity of a platform is by necessity the root of all our trust. We must know that from the beginning of the boot process the system is in a "known" state and that none of its "trusted" code, data, running processes, system components, or platform hardware have been altered. We must be able to reliably determine through a check that the platform has not been compromised.

3.4.1.2 Access Policy

We need to assure that code, data, running processes, and system resources are protected by access control policies. Access policies determine what entities can create, read, modify, execute, and change access to resources. The code that enforces policy, the rules that define policy and the linkage between them must be protected. Access policy is related to secure storage, which is a feature identified by TCPA and, for example, is a basis for SE Linux*. SE Linux implements mandatory security policies.[xiv]

3.4.1.3 Authenticity

Authenticity tells us where code or data come from. We can use this information to determine whether resources we receive over the net are to be accepted for use.

3.4.1.4 Trusted Path

Trusted path is a mechanism which assures that a user is directly interacting with trusted software. Is must not be possible for attack software to imitate trusted path.[xv]

3.4.1.5 Trusted Boot

A secure system at run-time can often be compromised at boot-time. For example, an attacker with a boot disk containing a small OS can use it to subvert an otherwise "trusted" system. (On the other hand, a client can use a boot disk to gain "trusted" access to an untrusted system.) A secure system must have a way of preventing boot-time attacks.

3.4.2 Operational Characteristics

In addition to capabilities, there are some operational characteristics which are important to evaluate.

3.4.2.1 Reliability of Security Implementation

This characteristic summarizes how confident we are in a particular product's implementation: quality of the security architecture and how well it is reviewed.

3.4.2.2 Breach Impact

This characteristic summarizes how well the product limits security breaches. For example, do breaches leave an entire system vulnerable or limit impact to isolated components.

3.4.2.3 Administration

This characteristic summarizes security administration. For example, who are the administrators and how much control do they have. This is a convenience, control, and trust issue.

4 Security Technologies

In this chapter we look at security methods and technologies available in the market now, or in the near future, that would allow the software implementation of SIM to achieve capabilities listed in Section 3.4: integrity, authentication, access control, trusted path, and trusted boot. Based on these technologies, we discuss best solutions in Chapter 5.

4.1 Software Methods

4.1.1 Tamper Resistant Software (TRS)

TRS is software which is resistant to observation and modification. It is a strong form of obfuscation.. It implies that in tamper resistant software it is difficult to observe and analyze the software to decide where a particular function is performed and how to change the software so that the desired code is changed without disabling the portion whose functionality the attacker wishes to retain.

4.1.1.1 Principles of TRS

Preventing operations on the secret component contained in the software is critical to ensuring that the application has not been tampered with. Aucsmith[xviii] mentions these principles based on the need to hide a secret in software and to ensure that recovery or alteration of that secret is difficult.

a. Disperse secrets in both time and space

Secrets should never exist in a single memory structure (cannot retrieve it by scanning active memory) and should not be processed in a single operation (cannot deduce by monitoring code execution)

b. Obfuscation of interleaved operations

The complete task to be performed by the software should be interleaved so that a little bit of each part of the task is performed in successive rounds of executing code. Additionally the actual execution should be obfuscated to prevent easy discovery of interleaved component results. Such obfuscation is accomplished by self-decrypting and self-modifying code.

c. Installation unique code

Each instance should contain unique elements that could be added at installation time in form of different code sequences or encryption keys.

d. Interlocking trust

The correct performance of a code sequence should be mutually dependent on the correct performance of many other code sequences.

4.1.1.2 Implementations - Medusa

The first implementation of TRS was done to protect a reference implementation of Common Desktop Security Architecture from modification. Later this was

used for content protection. This was a post compiler implementation that took native IA32 machine code as input and produced native IA32 self modifying code that self-decrypts and re-encrypts itself as it runs, exposing only a small portion of the code at any given time. However it did nothing to protect the data, which was still on the stack in the same place as the compiler had allocated it. This is useful to some internal attacks. This was an architecture-specific approach that did not meet the "author once, play everywhere" requirement. This also implied that self-modifying encrypted code was not a possible solution. A later implementation undertook the task of re-inventing TRS with additional constraints. It takes the philosophical approach of a compiler that would optimize obscurity much as we would attempt to optimize performance. The strategy is to build tools that introduce entropy at every level so as to require a substantial amount of human effort to analyze and attack software. This new source based approach is fairly complimentary to machine code-based, doing a good job of protecting data while struggling to protect code from meaningful modification. It provides new approaches for preventing easy analysis relating low-level operations to high-level ones, and offers some protection for dynamic analysis of instruction traces and simulations.

4.1.2 Virtual Machines (VMware*)

A Virtual Machine is a software environment that encapsulates one or more operating systems and applications that run inside or "under" the VM. The OS can't tell the difference between operating in a VM or in a "real" machine. Virtual machines are isolated entities.

VMware* (VMM) is a "hosted" type: it uses the host OS memory management, processor scheduling, hardware drivers, and resource management. Software running in the VM executes directly on the underlying processor; there is strong isolation between virtual machines based on CPU hardware privilege levels. VMM runs on the PC platform on a hosted system, coexisting with a host OS. The host OS provides the driver and resource management support and performs I/O operation on behalf of the guest OS.

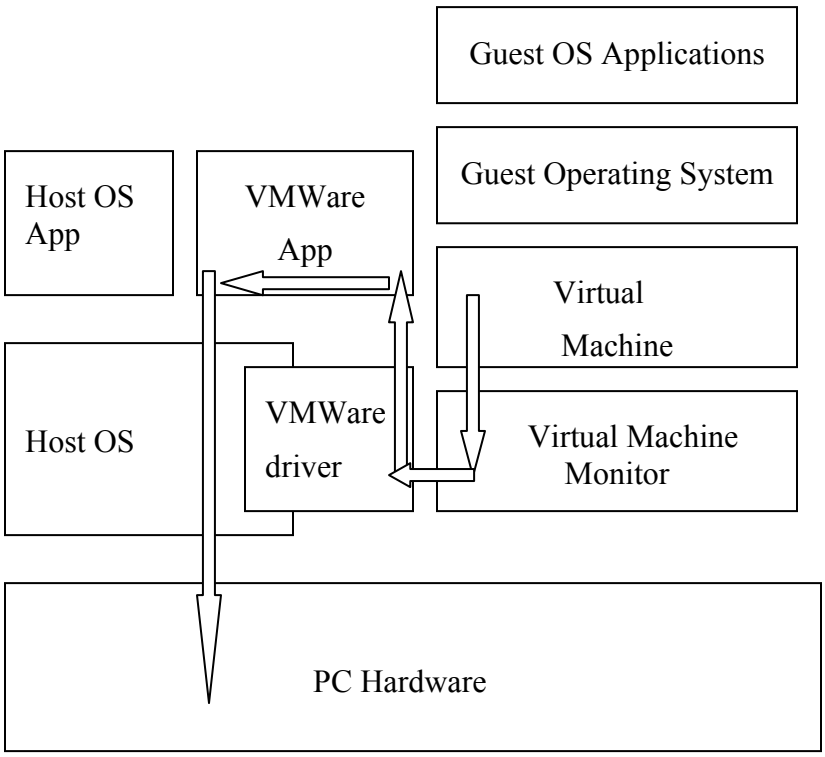


Figure 4. VMWare architecture

There are 3 products: Workstation, GSX server, and ESX server. We are looking at the workstation only. The workstation consists of three basic components: VMX driver that is installed within the OS to gain the high privilege levels, VMware application running in Ring 3, and component – VMM running in the kernel memory and having Ring 0 privileges.

VMware allows three different networking configurations:

1. Host-only network that exists only in the host OS and is used primarily for communication between the host and guest;
2. Bridged networking that multiplexes the host system's real Ethernet interface, so a guest system can talk on the host's network. It provides an abstraction for "virtual Ethernet hubs". This capability allows virtual machines to be interconnected in a conventional way.
3. NAT – host-only network that uses NAT to communicate with outside networks.

There is no mention on securing communications between VMs in VMware Workstation.

There was an attempt to use VMware for creating the NetTop technology by the National Security Agency for creating a system compatible with standards-based IT security solution[xix] products . The NSA assessment pointed out that VMware includes the

capability to copy and paste data between VMs via a clipboard, and that this feature does not include sufficient safeguards for use in high assurance systems.

4.2 Hardware Assisted Methods

Ultimately, using hardware for providing trusted boot, storage, and execution would be the best security solution. Currently this option is not available for PC or PDA platforms. In this paper, we restrict our consideration to only those means of hardware support that are either available or currently emerging in the market.

4.2.1 Removable security devices

Removable security devices are finding more use in modern computer systems. Although our solution does not recommend against using these, we are not basing our analysis on removable security devices, for the following reasons.

- We would like to use the benefits of high storage capabilities of the open platform systems;
- We would not like to depend on the availability of the removable device interfaces;
- When removable security devices are used with a laptop or PDA, some processing still occurs in common memory area, so these devices do not guarantee full protection. Thus some of the methods considered in this paper could be used in conjunction with these devices.

4.2.1.1 Smart Cards

One of the most popular hardware security solutions is a Smart Card. Use of one type of smart cards – SIM - was extensively analyzed in context of cellular networks (see Chapter 2). Other uses of Smart Cards include banking (Europe).

4.2.1.2 USB Tokens

A USB security token is a portable, stand-alone device that provides similar functions and features as smart cards: non-volatile secure storage, on-board cryptographic processing, RSA key operations, random number generation, and key generation.

Overall size of a USB token is about the size of the car key. Usually they are water resistant and come in a tamper evident casing.

4.2.2 Fixed Security Device: Trusted Platform Module

The Trusted Platform Module(TPM) is a key component for implementing trusted computing platforms as defined by the TCPA specification[xx]. It is a collection of hardware, firmware and/or software that support the following functions

- Algorithms: RSA, SHA-1, HMAC
- Random Number Generation
- Key Generation

- Self Tests

The TPM inherently provides a root of trust by making keys and data within it unreadable externally. It also provides a means for implementing secure storage using RSA key technology to encrypt data and keys. The functionality of the TPM can be used to seal data passed in, or private keys generated inside, the TPM. The sealing operation encrypts not only the data, but also TPM specific register values, which are unique for that TPM. This data or private keys, which are internally generated, are still secure when stored outside the TPM.

To unseal the data, three conditions must exist:

1) the appropriate key must be available for unseal

2) the TPM specific registers must be the same values that existed at the time of the seal operation

3) the TPM proof value must be the same as that encrypted during the seal operation.

By requiring the TPM register values to be duplicated at unseal and the TPM proof value to be checked, the seal operation allows software to explicitly state the future "trusted" configuration that the platform must be in for the decrypted key to be used and for decrypt to only occur on the specified TPM.

A number of key types are defined within the TPM. Keys may be migratable or non-migratable. A migratable key is a key that may be transported outside the specific TPM.

A non-migratable key is a key that cannot be transported outside a specific TPM.

The Endorsement key pair, is an asymmetric key pair, which is generated by or inserted in a TPM that is used as proof that a TPM is a genuine TPM. Each TPM is identified and validated by its Endorsement Key. A TPM has only one endorsement key pair and is the basis for the root of trust.

The Storage Root key (SRK), is the root key of a hierarchy of keys associated with a TPM; it is generated within a TPM and is a non-migratable key. Each TPM contains a SRK, generated by the TPM at the request of the Owner. Under that SRK are two trees: one dealing with migratable data and the other dealing with non-migratable data.

Signing keys must be a leaf of the Storage Root Key hierarchy. The private key of the key pair is used for signing operations only. Storage keys, are used to RSA encrypt and RSA decrypt other keys in the Protected Storage hierarchy, only. Identity Keys, are used for operations that require a TPM identity, only.

Binding keys are used for TPM Unbind operations only. A bind operation (performed outside the TPM) associates identification and authentication data with a particular data set and the entire data blob is encrypted outside the TPM using a binding key, which is an RSA key. The TPM Unbind operation uses a private key stored in the TPM to decrypt the blob so that the data (often a key pair) stored in the blob may be used.

The TPM provides startup self-tests and a mechanism to allow self-tests to be run on demand. These ensure the functionality of the RNG, the integrity registers, the

Endorsement key pair integrity, the RSA sign and verify engine, and the integrity of the protected capabilities of the TPM. When the TPM detects a failure during any self-test, the part experiencing the failure will enter a shutdown mode and an error code is returned. Registers are written with a fixed set of data as defined by the manufacturer.

The TPM identification and authentication capability is used to authenticate an entity owner and to authorize use of an entity. The identification and authentication data for the TPM Owner and the owner of the Storage Root Key are held within the TPM itself. The identification and authentication data for other owners of entities are held and protected with the entity. The identification and authentication protocols use a random nonce provided by the TPM to prevent replay attacks and man-in-the-middle attacks. Access control is enforced in the TPM on all data and operations performed on that data.

4.2.3 Interfaces to TPM

There are three basic interfaces to TPM: CAPI, PKCS11, and TSS.

4.2.3.1 CAPI

Microsoft Crypto API is a framework that supports various security services on the Windows* platform. Among those supported services are encryption, key generation, random number generation, and others. CAPI allows the use of third-party cryptographic algorithms from a cryptographic service provider (CSP). Through this interface, application software is able to make use of encryption algorithms by selecting them at runtime.

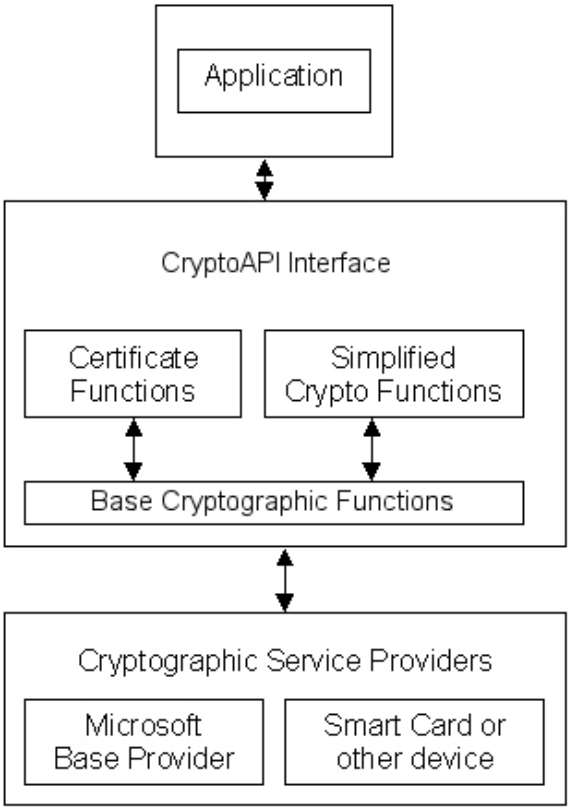


Figure 5 CAPI architecture (http://msdn.microsoft.com)

Default CSP set of functionality provided by Microsoft is RSA Base Provider. The RSA public-key cipher is used for both key exchange and digital signatures, with a key length of 512 bits. The RC2 and RC4 encryption algorithms are implemented with a key length of 40 bits. The MD2, MD5, and SHA hashing algorithms are also provided.

More details can be found at Microsoft site (http://msdn.microsoft.com).

4.2.3.2 PKCS

The PKCS11 API for C standard specifies an application-programming interface called Cryptoki to devices that hold cryptographic information and perform cryptographic functions. Cryptoki (short for Cryptographic Token Interface) follows a simple object-based approach, addressing the goals of technology independence (in any kind of device), resource sharing (in multiple applications accessing multiple devices) and presenting to applications a common, logical view of the device called a cryptographic token.

The main objects in the PKCS #11 API are Slots, Tokens and PKCS11 objects. Token is a general term for devices, which hold cryptographic information (PKCS11 objects like keys or certificates) and perform cryptographic functions (like digital signatures, random number generation or encryption) after having opened a session. A slot is a container, which can potentially hold a token.

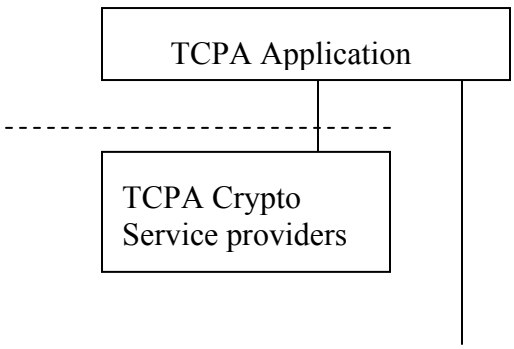
The PKCS11 functionality is split roughly into three parts:

- Administrative operations: functions like login, open Session, etc.
- Object management operations: createObject, destroy, etc. and functions to synchronize the content of the objects in memory with the card
- Cryptographic operations: sign, digest, encrypt

4.2.3.3 TSS

TCPA Software Stack (TSS) that is an integral part of each platform, and provides functions that can be used by enhanced operating systems and applications. The TCPA Software Stack (the TSS) is the supporting software on the platform supporting the platform's TPM.

Figure 6.



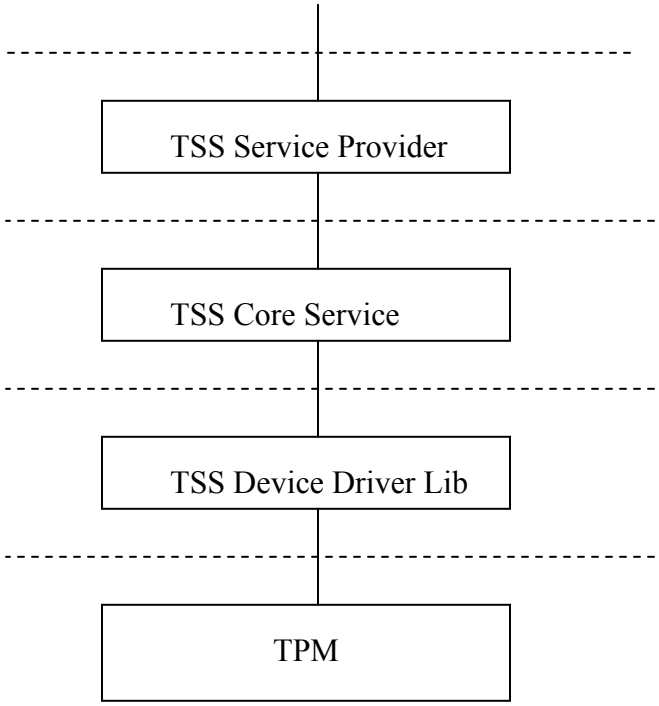


Figure 6. TSS architecture (www.trustedcomputing.org). Dotted lines represent the interfaces.

The TCPA Device Driver Library (TDDL) provides two functions: a standard interface for TPM (TDDLI) and transition between the User Mode and Kernel Mode. There will typically be one executable image of each of these per TPM on the platform.

The TSS Core Services (TCS) resides in User Mode and communicates to the TPM via the TPM Device Drivers Library Interface (TDDLI) provided by the TDDL. There will typically be one image of this component per platform and it executes as a system service. This module provides all the primitives and more sophisticated functions such as key management required to efficiently manage the TPM's limited resources. The interface to the TCS is the TSS Core Service Interface (TCSI). It provides controlling and requesting services from the TPM.

TSS Service Providers (TSP) are the top-most modules and provide a rich, object-oriented interface for applications. The interface used by the applications to access the TSP is the TSS Service Provider Interface (TSPI). While not an architectural requirement, it is intended that the TSP can obtain many TCPA services such as TPM byte stream generation, key management, etc from the TCS.

Another type of module that may make use of the TCS is an RPC server. This module marshals the TCS functions and data from the TCPA platform to another platform or device.

None of the TSS modules, components or the RPC communications affect the trusted properties of the TPM. All modules, components and interfaces outside the TPM are considered untrusted in relation to the TPM.

More details can be found at www.trustedcomputing.org .

4.2.4 Two Potential Uses of TPM for Software SIM Implementation

TPM is defined in the TCPA (Trusted Computing Platform Alliance) specification. Section 4.2.2 above follows the description of TPM as presented in the specification. TCPA defines procedures for verifying platform integrity and trusted boot using TPM. Obviously, just having TPM on an existing platform does not provide the desired functionality. The BIOS has to undergo a significant rearchitecture, major changes to OS loader and other parts of the OS have to be done. With all the required support from BIOS and OS, TPM could provide boot integrity and measure the platform integrity. This would provide a significantly higher level of protection for the data and processes we care about (see Chapter 3.2).

Another option is to use TPM as a security token, completely out of the TCPA context. In this case, we can view it as a small device that provides an isolated execution environment and a limited storage capacity. The best fit to implementing Virtual SIM would be to run the GSM algorithms inside the TPM. However, the set of algorithms that is provided in TPM is completely different (see Section 4.2.2 above). Thus, we can use TPM for sealing the data for storage on the hard disk, as well as for limited storage inside TPM. Other use of TPM would be assist in creating secure channels, with another "island" of isolated execution environment on the other end of the channel (e.g. smart card or a server on the operator's network).

4.3 Combined TPM and TRS Solution

TPM (see Section 4.2 above) and Tamper Resistant Software methods (see Section 4.1.1) provide the complementary set of security features and thus can be used in conjunction. Used together, they provide better security of storage for the shared key and more secure execution environment, so that the keys are not in the open during execution time. TPM can be used to store the secret shared key so that only restricted applications can access it using the TPM identification and authentication capability. Currently we can use TPM to encrypt and store our shared key but when the key is being used - it has to be decrypted by the TPM and sent in clear to the application. This is a potential security hole that will be fixed in the later versions.

Another potential hole is the problem of securely provisioning the keys into the TPM. TPM can also provide trusted boot with some OS modifications. This ensures that we have taken care of storing and accessing the shared key and also the state of the system. Now we need to take care of protecting the key while it is being used and also the algorithms since GSM algorithms are not published, and for this we use the TRS technology. This ensures that the key and the algorithms are dispersed in time and space so that any application snooping it cannot get hold of the key. Currently Intel's TRS technology is based on asymmetric RSA keys and the fact that they can be used in part

and that you don't need the whole key at a given time. This is an aspect that needs to be looked at as GSM uses symmetric key.

Tamper-Resistant Software methods create some degree of protection for the execution environment, storage, and boot. These are fortified by advantages offered by TPM: key storage / generation, authentication, etc.. The OS this solution is running on may provide an independent, added layer of protection (e.g., SE Linux's Access Control). TRS + TPM solution does not depend on it.

4.4 Policy and Implementation Aids

Software design and implementation play complementary roles to software and hardware technologies described above. In this section we address some policy and implementation heuristics that fall outside the realm of specific technologies. For example, software SIM places constraints on a system that are not typical for consumer PCs. The security goal for most systems is to assure that it is secure from unauthorized third party access. A new wrinkle that SIM introduces is that third party data and communications must also be secure from the system owner.

International embassies are a useful analogy for modeling policy. The host country determines which embassies are allowed in their territory and under what conditions an embassy is removed. Both embassy and host country have security expectations. Mechanisms and policies are put in place to protect embassy information from the host and other embassies. The host also has mechanisms and policies to protect its information from invited embassies.

In a similar manner, we must create policies so that a device owner can install, uninstall, virus-scan, and firewall all communications. The owner cannot observe secrets or manipulate programs in the subsystem that is the third-party's trusted computing base (TCB). The TCB allows the subsystem to sign and encrypt communications and verify the integrity of incoming programs and messages.

In addition to the security policy just described a few implementation guidelines are rules of thumb that help build secure systems.

1. Security must be encapsulated in as small a kernel as possible (code size and API width). It is easier to implement small pieces of code correctly.
2. Security must be verified (or minimally validated) from the top down to the hardware. Evaluating a single component is not enough because a security hole in any layer may compromise an entire system.
3. All kernel source code must be available for inspection. Evaluators with fresh perspectives, experience, and independence can locate overlooked problems. Independent analysis and inspection are required to raise confidence and promote third party buy-in.
4. Be careful about where secrets are stored and how they may be probed at run-time. Attacks can take advantage of statistical analysis to locate critical keys. Information left in registers or temporary storage can provide valuable clues.

5 Solutions for Software SIM Implementation

The trust requirements captured in Chapter 3 guide us toward solutions with access control as one of the major pre-requisites. Access controls in today's operating systems are implemented using technologies focused at various clients. At the user level, access controls are implemented using account domains, which partition the computing systems into multi-user environments. Each user has a username-based account protected by a password and an access policy based on ACLs (Access Control Lists) for resources such as files, processes, and network. A refinement of ACLs uses a system of derivative rights. For example, applications can be assigned rights that determine at a fine grain level which resources they can manipulate. Further control is possible by defining classes of clients with restricted or enhanced access control.

In this chapter we will use the capabilities derived in Section 3.4 to evaluate several classes of solutions. We will look at solutions that can be built on readily available and known software platforms. We partition solutions into three classes based on the degree of execution isolation.

- Windows 98 for example offers very little isolation between users. Users are only separated by profile data while the file systems are not protected. These systems are analyzed in section 5.1.
- Linux, UNIX and Windows XP (assuming NTFS usage) do provide file system protections at the user level. They offer discretionary control where a user in possession of a resource controls its access rights. SE Linux provides better isolation than standard UNIX and XP by using mandatory access controls. These systems are analyzed in section 5.2.
- Virtual machines provide another level of isolation. It is possible to run entire operating systems within each virtual machine, thus providing good isolation and crash proof protection. These systems are analyzed in section 5.3.

Our analysis will be based on how well software implementations of SIM within each of these solution areas stand up to hardware SIMs. It must be recognized that current GSM systems do not meet the requirements presented in Chapter 3. These requirements must be viewed as the goals for GSM and any software SIM solution we propose. There are multiple faults discovered in GSM systems (see Section 2.1.4), and, although 3G networks are doing a much better job, they are also far from being perfect. Therefore, for each requirement we need the same level of strength as that provided by current GSM systems. There is no need to make the terminal security stronger than the level that the air interface or the network provides – the overall system security is only as strong as its weakest link. In other words, the achievable level of security that we can expect from our implementation of SIM in software on the open platform should be comparable to the level of security that is currently accepted in the industry.

Our analysis is based on costs and benefits:

- **Solution Cost** – Here we look at development resources, in particular engineering effort.

- **Attack Resistance** – It is not our goal to create a completely secure system. A watertight solution is just too expensive and not necessary. Our goal is to make attacks difficult enough such that their cost is higher than the value they return. Attack resistance is one dimension of cost benefit from the attacker's point of view.
- **Potential Attacker Gains** – This is the second dimension to attacker's cost benefit. Together with attack resistance, attacker gains give us a sense of whether an attacker is rewarded for their efforts.
- **Attack Detection** – A successful attack that is not detected is particularly problematic. Early detection can limit losses and give engineers an opportunity to fill security holes.
- **Attack Recovery** – A quick recovery limits denial of service attack.
- **Bottom Line** – The bottom line documents our key learning about this solution set.

5.1 Architectures Without Isolated Environments

One starting point is an open PC platform with standard hardware and software that does not isolate environments. Examples include Microsoft Windows systems for home use (Windows 98, ME, and XP Home). They are marketed to non-hostile environments where ease-of-use has precedence over security. For example, FAT32 in Windows 98 and ME does not support file access controls. Even XP Home, which uses XP Professional as a base, has simplified security.

Microsoft introduced isolated memory between running processes with the arrival of Win32. However, this protection focuses on protecting processes against disruption by faulty neighbor processes and not hostile intentions. For example, any user can overcome memory protection by replacing memory drivers.

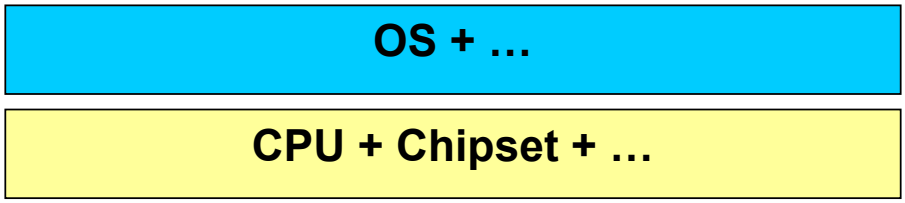


Figure 7 SIM with system without isolated environments

In the following tables we will assess the security of platforms without isolated environments based on the criteria identified in section 3.4. Simultaneously, we will evaluate any trust enhancements to the hardware such as TPM or smart cards and software techniques such as tamper resistant software.

5.1.1 Operating System Summary

	Windows 98	XP Home Option 1 FAT with default users
Integrity	No off-the-shelf support.	The home edition has simplified XP professional security. In one typical installation each interactive user has administrative access giving them full control. Any user with administrative rights will be able to change the system without detection.[2]
Access Policy	Limited access policies are available off-the-shelf. Critical system files may be set for limited read and write protection. Any user can set them. Additional enhancements are required, for example, file encryption may be used to further limit read and write access. TPM can store critical keys.	When the FAT16/32 file system is installed, there is no file protection
Authenticity	Off-the-shelf authentication support for code download.	Same as Windows 98.
Trusted Path	No off-the-shelf support. The trusted path problem cannot be solved easily. This breaks everything because any user can initiate a man-in-the-middle attack. We will never know whether information is being intercepted and modified.	Same as Windows 98.
Trusted boot	Not available and not easily added	Same as Windows 98.

[2] XP Professional and Home share the same NT based kernel. Professional is a superset of Home.

	Windows 98	XP Home Option 1 FAT with default users
Reliability of Security Implementation	Off-the-shelf reliability is based on Microsoft specification and evaluation. The system would require extensive and expensive modifications to make it viable. Minor implementation oversights could create security holes that are easy to attack and make the system unstable.	Same as Windows 98.
Breach impact	It is not difficult to completely compromise or disable an off-the-shelf installation.	Same as Windows 98.
Administration	Off-the-shelf, any user can administer the system.	Same as Windows 98.

5.1.2 Analysis of Operating Systems with Security Enhancements

In the previous section we summarized the security capabilities of OSs with limited isolation. In this section we analyze the costs and benefits of enhancing them with technologies described in Chapter 4.

1. Off-the-shelf Windows 98 or XP Home.
2. Above plus encryption, tamper resistance, and signing
3. Above plus TPM. (See section 4.2.4)

    a. TPM as a security token, that we view as a small device that provides an isolated execution environment and a limited storage capacity.

    b. TPM following TCPA specifications. Assumes significant modifications to the BIOS, OS loader, and other OS components to provide boot integrity and platform integrity.

c.

	1. Off-the-Shelf	**2. Software only Customization**: Encryption, tamper resistance[xvi], signing	**3. Added hardware Customization**: **a. TPM only** **b. TPM with BIOS and OS support for TCPA**
Solution cost	Low	High - extensive system modifications required by highly skilled engineers. Difficult to validate improvements.	a. High – Same as software only. b. Higher than software only. Substantial effort would be required to take full advantage of the TPM.
Attack resistance	Low	Depends on the solution. Easy attacks will take substantial effort to prevent.	a. The difficulty of preventing man-in-the-middle attacks will quickly erase any hardware gains. b. The TPM can be used to determine the integrity of critical components during system start-up. This can serve as a warning of an attack on sensitive algorithms and data.
Potential attacker gains	An attacker may be able to copy the subscriber's private key and clone the subscriber's identity.	Same as off-the-shelf.	a., b. Same as off-the-shelf.

	1. Off-the-Shelf	2. Software only Customization: Encryption, tamper resistance[xvi], signing	3. Added hardware Customization: a. TPM only b. TPM with BIOS and OS support for TCPA
Attack detection	None	A write protected boot disk could check for compromised components.	a. Not much better than software customization. b. A TPM may be able to detect problems at boot-time. Run-time checks require a trusted challenger between two endpoints (e.g., TPM to dongle with smart card or TPM to trusted network server). The installed OS is susceptible to man-in-the-middle attacks and thus cannot be trusted to report accurately from the TPM to the software running locally.
Attack recovery	Reinstall; after that, device would have to be re-provisioned.	A write protected boot disk and CD could replace compromised components. System rebuild may be necessary to remove Trojan Horses.	a., b. Same as software only. If the TPM is compromised, recovery will require a service call with the possible replacement of the TPM.
Bottom Line	Not viable	Upgrade to XP Professional and go to next section. It will cost less than, be more reliable, more secure, and easier to use than modifying Windows 98 or XP Home.	a. Adding trusted hardware to systems in this category has little value. b. A TPM could detect a good boot but without a run-time trusted path after start-up it does no good.

5.2 Architectures With Isolated User Environments

The second beginning architecture is a typical multi-user environment such as UNIX, SE Linux, or XP professional (Windows NT derivations). They support resource control out of the box, for example, file access policy and isolation between running processes. Our goal is to determine where their weak points are and whether there are software and hardware enhancements that can improve them.

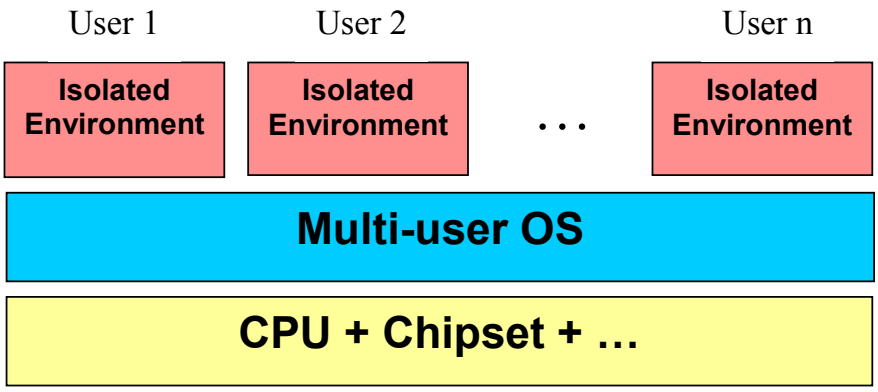


Figure 8 SIM with system with virtual machines supporting isolated environments

XP Home option 2 uses NTFS and "Limited" accounts for all but the administrator. File access is controlled through personal and shared folders. The question is - why not just use XP Professional, which is a superset of the home edition? The professional edition can be made to look just like the home edition plus it supports encrypting the file system and uses credentials to give third parties file access. In addition, it supports access control lists, administrator defined security policies for users, certificate services, and software restriction polices based on signed code.

5.2.1 Operating System Summary

	SE Linux	Linux	XP Professional
Integrity	Off-the-shelf support sufficient to implement integrity. For example, critical components could be placed in isolated user accounts with limited access. Implementation effort is required to monitor integrity. For example, specialized processes could be constructed to monitor components for change.	Same as SE Linux	Same as SE Linux.
Access Policy	Off-the-shelf support is a strong point for SE-Linux. It has fine-grained control using types, roles, and identities to protect files, directories, sockets, IPC. SE Linux supports *mandatory* security policy. Security attributes are under the control of the system.[3]	Off-the-shelf support is probably sufficient. It has coarse-grained control (super user vs. user) on limited types (files). Security support is *discretionary*. Ordinary users may assign security attributes.	Off-the-shelf support is probably sufficient. It has coarse-grained control on files and supports access control lists. This assumes a NTFS installation. Security support is *discretionary*. Ordinary users may assign security attributes. In addition there is support for encryption with certificates that can be used by third parties to decrypt. Certificates are self-signed – bypassing the need for a certificate authority.

[3] Mandatory Access Control: An ordinary user may not be able to modify access rights to documents in their possession. For example, Helen may be able to read a document but not give read access to Eric. In addition, the concept of a "super" user who has every access right to all documents is not supported.

	SE Linux	Linux	XP Professional
Authenticity	No off-the-shelf support.	Same as SE Linux	Off-the-shelf support for code download authentication and software restriction policies base on signed code.
Trusted Path	There are trusted paths if the administrator sets up isolated carefully managed user accounts. Trusted path is broken in accounts that have unknown software downloaded into them.	Same as SE Linux	Same as SE Linux
Trusted boot	Not off-the-shelf.	Same as SE Linux	Same as SE Linux
Reliability of Security Implementation	Specification and evaluation overseen by the NSA. Small kernel makes evaluation easier.	Specification and evaluation is ad-hoc – by interested parties. Large kernel makes evaluation difficult. (Free BSD has a small kernel and a better evaluation procedure.)	Specification and evaluation overseen by Microsoft. Large Kernel makes evaluation difficult.
Breach impact	Breaches in security can be limited to a bounded domain. For example, breaching the send email program only exposes email to attack. Furthermore, programs do not inherit access from their user. For example, a web browser will not have all the rights of an administrative user when run by an administrator.	If an attacker gains access to root through a defective program or other means, the whole system is compromised	Same as Linux (except its called administrative user).

	SE Linux	Linux	XP Professional
Administration	We have fine-grained control of what roles users (including administrators) can take	Who gets root access – operator or subscriber? If it's the subscriber, the security of operator information is compromised. If it's the operator, the subscriber may not be happy with their level of control	Same as Linux

5.2.2 Analysis of Operating Systems with Security Enhancements

In the previous section we summarized the security capabilities of a several OSs that support isolated environments. In this section we analyze the costs and benefits of enhancing them with technologies described in Chapter 4.

1. Off-the-shelf XP Professional, Linux, or SE Linux.
2. Above plus encryption, tamper resistance, and signing
3. Above plus TPM. (See section 4.2.4)

    a. TPM as a security token, that we view as a small device that provides an isolated execution environment and a limited storage capacity.

    b. TMP following TCPA specifications. Assumes high significant modifications to the BIOS, OS loader, and other OS components to provide boot integrity and platform integrity.

	1. Off-the-Shelf	**2. Software only Customized**: signing, authentication, TRS	**3. Added hardware Customization**: TPM **a. TPM only** **b. TPM with BIOS and OS support for TCPA**
Solution cost	Low Digital signing methods could be implemented and integrated into the system	Low for authentication (signing) and integrity (background monitoring process). Moderate to high for TRS. It depends on how much of the software must be protected by TRS techniques.	a. Low cost for storing a limited set of critical keys. b. Moderate/high cost for extending protected store. High cost for boot integrity and runtime integrity (BIOS and OS modifications). Depends in part on access to OS internals (Linux vs. XP). TPM costs under $5.
Attack resistance	Moderate to high for well set up systems. SE-Linux will limit breaches. Low resistance for boot time attacks.	Somewhat better than off-the-shelf. Tamper resistance software can be used to limit breaches and make gains harder fought.	a. Somewhat better than software customized. b. Higher resistance to system component modification (better integrity monitoring). Better protection of critical keys. In case of SE-Linux source, we can take full advantage of the TPM to extend trust to the entire kernel from boot onward. To achieve high resistance, all software components must be authenticated, and access control has to be used correctly.
Potential attacker gains	An attacker may be able to copy the subscriber's private key and clone the subscriber's identity.	Same as off-the-shelf.	a., b. Same as off-the-shelf.

	1. Off-the-Shelf	2. Software only **Customized**: signing, authentication, TRS	3. Added hardware **Customization**: TPM **a. TPM only** **b. TPM with BIOS and OS support for TCPA**
Attack detection	None.	Custom isolated processes can monitor the system integrity. Attack detection is relatively easy to bypass.	a. Attack detection is relatively easy to bypass. b. TPM can monitor system integrity in a manner difficult to bypass.
Attack recovery	Depends on how limited the attack is. With SE Linux may limit the extent to which components may need to be rebuilt. XP has a system restore feature	Same as off-the-shelf	a. Same as off-the-shelf. b. If the TPM is compromised, recovery may need TPM to be replaced and will require a service call.
Bottom Line	SE Linux off-the-shelf will give us a lot. It is free. Users may prefer XP Professional.	TRS appear to come at a high price for their value.	a. TPM can be used as a secure store for some critical data. b. With substantial work and careful restrictions, this case provided strong protection. XP presents a problem here because we are not easily able to modify the OS.

5.3 Architectures With Virtual Machines Supporting Isolated Environments (VMware)

Here we consider architectures that are based on virtual machines running isolated operating systems. It is out of the scope of this document to consider hardware-assisted solutions with hardware-assisted isolated execution environments. Therefore, we'll limit our scope to software implementation of virtual machines with some hardware assist such as TPM.

Out of several implementations of virtual machines, VMware Workstation (version 3) discussed in Section 4.1.2 is our only obvious choice, since it is the most mature software virtual machine tool available for PCs.

There are several options for using it, depending on the choice of host and guest operating system and other methods used for improving security.

We choose Windows XP (Professional) as an important commercially available OS and open source SE Linux OS. We've chosen to consider these two operating systems, with clear understanding that we only cover a limited combination of the variety of operating systems, but knowing that they are the most interesting ones in the context of our analysis. We have four possible combinations:

1. Windows XP host, Windows XP guest.

2. Windows XP host, SE Linux guest.

3. SE Linux host, Windows XP guest.

4. SE Linux host, SE Linux guest.

Options 1 and 2 utilize isolation between two virtual machines provided by VMware. However, as noted in Section 4.1.2, this isolation is not strong. Security levels for XP Professional and SE Linux are comparable (see more detailed analysis in Section 5.2.2). Use of TPM module if it is available on the platform is limited by CAPI or TSS interfaces.

Options 3 and 4 have an advantage of an open source and thus modifiable host OS. If there is a TPM module on the platform (see Section 4.2.2), it is possible to modify the SE Linux kernel, so it supports TCPA integrity metrics and verification. Other significant efforts required to achieve this are creating the TPM driver for Linux and re-architecting BIOS. Overall, the engineering effort for implementing a TCPA compliant solution in SE Linux is well above the reasonable target for our software SIM implementation.

Another problem with using SE Linux as a host is that it is not as popular as Windows, a suitable OS for broad distribution, and we do not see it as a viable commercial option.

Overall, for the reasons above, options 3 and 4 are not suitable for software SIM solution.

This leaves us with a choice of option 1 or 2 above – the two are very similar and are considered together.

5.3.1 VMware Summary

VMware: XP as host OS, XP or SE Linux as guest OS

	VMware: XP as host OS, XP or SE Linux as guest OS
Integrity	Critical components can be kept in an isolated guest OS. No other processes are running there. ME and SIM processes can be both running in isolation. However, communication between VMs is not secure. Overall integrity comparable or somewhat better than XP.
Access Policy	As strong as the OS access control.
Authenticity	No value added by VMware. Windows XP provides some code download authentication. Digital signing methods must be implemented and integrated into the system
Trusted Path	VMware provides some degree of process isolation. However, communications between VMs are not sufficiently secure.
Trusted boot	Not possible without TPM and OS support
Reliability	Almost up to the level of commercial OS products.
Breach impact	There is an option in VMware to make all changes to guest VM transient. Not sure if it can be used to reduce the breach impact.
Administration	Who administers VMware determines security of its virtual machines

5.3.2 Analysis of VMware with Security Enhancements

In the previous section we summarized the security capabilities of VMware in conjunction with several OSs. In this section we analyze the costs and benefits of enhancing them with technologies described in Chapter 4.

1. VMware off-the-shelf with Windows XP as a host OS and Windows XP or SE Linux as a guest OS.
2. Above, plus software security enhancements: TRS, encryption, etc.
3. Above, plus TPM. (See section 4.2.4)
    a. TPM as a security token, that we view as a small device that provides an isolated execution environment and a limited storage capacity.

b. TPM following TCPA specifications. Assumes high significant modifications to the BIOS, OS loader, and other OS components to provide boot integrity and platform integrity.

	1. VMware with SE Linux as guest OS ("off-the-shelf")	**2. VMware with software customizations**	**3. VMware plus TPM:** **a. TPM only** **b. TPM with BIOS and OS support for TCPA**
Solution cost	VMware Workstation cost - ~$300.	Use of TRS methods increases cost significantly.	a. Moderate cost increase (TPM cast under $5). b. High cost: engineering effort to modify OS (SE Linux) and BIOS, to add/enhance TSS or other middleware to interface TPM (not fully available now); to implement Linux driver for TPM.
Attack resistance	VMware is not too strong. Problem areas: execution, storage, boot - man-in-the-middle attack is possible.	Somewhat better run-time and storage security. Somewhat stronger protection for man-in-the-midle attack.	a. Somewhat better storage protection: both limited storage inside TPM and use of sealing TPM mechanism. Somewhat stronger for man-in-the-middle attack. b. Significantly better due to system integrity measurement capability and trusted boot. Denial of service attack is still possible.
Potential attacker gains	In case of successful attack, Identity and SIM algorithms can be stolen, service can be cloned.	Same as in Column 1.	Same as in Column 1.
Attack detection	None.		a. None. b. TPM has capability to verify its registers and logs and detect if integrity was violated. Still, no detection at run-time.

	1. VMware with SE Linux as guest OS ("off-the-shelf")	2. VMware with software customizations	3. VMware plus TPM: a. TPM only b. TPM with BIOS and OS support for TCPA
Attack recovery	There is an option in VMware to make all changes to guest VM transient. If this option has to be used, device would have to be re-provisioned.	Guest OS can be restored (see 1st column); compromised components of TRS protected software may be replaced without OS re-install (depending on the type of the attack).	a, b. If TPM is not compromised, - same as in column 2. However, a new TPM may be required if the TPM root of trust is compromised as a result of the attack.
Bottom Line	VMware does not provide sufficient security benefits that would justify its high cost and usability issues.	Better security, but higher cost.	a. Better security, but still not enough to justify cost. b. Much better security. However it is achieved due to TCPA, not VMware.

6 Conclusions

We looked at the security requirements of GSM and 3G networks and the trust parameters for SIM. Exploring the available options, we came to the conclusion that the security provided by the closed platform is hard to achieve on the open platform. However we provide some solutions based on the currently available platforms, operating systems and tools. We think this is the best that can be done in the current environment.

6.1 Criteria

We rank a range of solutions roughly based on their level of security using our analysis from Chapter 5. The choice of any given solution will depend on market requirements. We have selected three criteria that guide a market-focused decision.

- *Security level.* This criterion focuses on what environment and what security enhancements, both hardware and software, are used. We partition this criterion into a) technologies that enhance security and b) environment that includes operation systems and virtual machines:
    a) *Security Enhancements*: One criterion is whether a solution takes advantage of TRS (Tamper Resistant Software) or TPM technology. Choice of TPM out of the variety of hardware tokens may be of interest because it lies on an important Intel roadmap even though other components in the system are not sufficiently mature to justify it.
    b) *OS / VM* : A second part of this criterion is the operating system and virtual machine environment. Again, in addition to security support, roadmaps and support play an important role it selection. For example, XP and SE Linux provide differing levels of security but also live on different roadmaps. Virtual solutions, such as VMware, are also of interest because of the role virtual machines are expected to play in some security roadmaps. The environment also determines solution audience: for example, XP or Linux have value in different domains.
- *Administration*: System administration determines how effectively any software and hardware technology is utilized. A poor choice of administrative policy will invalidate technical gains. Policies have a direct impact on a user's and service provider's sense of privacy and control. Who administers a given component is a critical decision metric. It determines data, communication, and process access.
- *Cost*: There are several cost factors. Apart from engineering development, quality assurance and related business costs, there are costs related to protection of investment. For SIM the cost is based on the fraudulent use of the network, which is the SIM is intended to prevent. This is mostly calculated with hindsight starting with some initial estimates based on the security strength of the solution.

In the following section we look at some of the cost trade-offs for implementation.

6.2 Cost Tradeoffs

Below are costs for the software components. Keep in mind that some solutions require a combination of components. An example is VMware, which may include XP and Linux as host and guest operating systems.

Component	Cost
XP Home	$200
XP Professional	$300
SE Linux	Free download
VMware	$300

In addition to component costs there are learning and development costs for various solutions.

- *XP*: Because XP is well established in the community, software engineers generally know how to use it and IT professionals know how to configure it correctly. However, cost of modifying it by third parties to take full advantage of TPM is probably prohibitively high. In particular, the task of enhancing XP to do integrity metrics at boot and run time is out of the scope of many projects.
- *SE Linux*: Linux is also well known but SE Linux adds complexity. In particular, the access control mechanisms in SE Linux are not easily mastered. The cost to modify it is within a team's reach but probably high. It is modifiable because its source code is available. This makes it possible to take full advantage of TPM during boot and run-time integrity checks. SE Linux installation is a two-step process, which makes it slightly more difficult to administrate. (One step is to install Linux and the second is to install the SE Linux enhancements over it.)
- *TPM*: Hardware cost for TPM is under $5. There are also software costs associated with TPM depending on how it is utilized. For example, if TPM has to be used under SE Linux, drivers have to be written. In case of Windows XP drivers have to be evaluated to determine whether they provide the required protection. Using TPM for integrity metrics in XP is more costly: it involves re-architecture of BIOS and changes to OS.
- *TRS*: Using tamper resistant software is labor intensive and requires substantial expertise for even small subsystems. Much of its implementation must be hand crafted and simple changes are costly. It does not scale well making it impractical for large-scale use. Its use has to be carefully targeted.

6.3 Administrative Tradeoffs

When we choose a particular combination of environment and enhancement technologies we also must choose how and who will administer those environments. The choice of administrative policy is as important as the technology. Poor policy choices will undo

many hardware and software safeguards. There are potential conflicts of interest between system users and service providers over the access rights that ultimately define who has control over the system. For example, many users feel that administrative control of a device they own is critical. Users want to have unrestricted ability to install software applications that meet their needs. On the other hand, service providers are resistant to supporting any device that is out of their control. They are concerned with the theft or disablement of services that form their revenue stream.

- *Windows XP*: An expectation of XP users is that many of them will have local (and sometimes remote) administrative control. Much of the software written for MS Windows assumes an administrative free hand. The problem is that anyone with administrative control has unrestricted access to data, processes, and communication. However, placing administrative control in the hands of a trusted third party changes the equation. We can construct a system that is not trivial to attack by users or third parties. This makes XP, even without hardware or software enhancements, an interesting environment. A question is whether system owners (users) would accept administration by a third party, pay for the third party support, accept strong restrictions on approved software, and restrictions to web and mail activity. Using administration to solve security issues moves most of our costs to infrastructure and behavior. However, we then end up with laptops that are just as inflexible as traditional cell phones. This solution underscores the importance of TPM. Even though TPM incurs up front hardware and software cost, it provides solutions that have more acceptable user models.
- *SE Linux*: Linux has a long tradition of partitioning users into user and root access. Users do not necessary expect to have root (administrative) access. Typical applications run in user mode. In addition, SE Linux has been enhanced such that root users do not have unrestricted access to user documents in SE Linux (NOTE – our analysis is not complete here – follow up is required to determine what root users can do). However, who administers the system remains a critical question. Will system owners accept and pay for third party administration?
- *VMware*: VMware provides substantial administrative flexibility with significant added complexity. The question becomes who administrates the VMware component plus each of the operating systems (hosts and guests). For example, a third party may administer VMware plus the guest virtual machine and the owner could administer a host OS. This flexibility adds complexity because there may be multiple OSs in addition to VMware to maintain. This added complexity makes a security oversight easier to get through.

6.4 Environment with Security Enhancement Tradeoffs

Figure 9 is a matrix of the security enhancements and operating system and virtual machine environments. We have restricted our solutions to three environments of interest: XP Professional, SE Linux, and VMware with Windows XP as a host OS and XP or SE Linux as guest OS. TPM and TRS are the chosen security technologies used to

enhance these operating systems. For example, cell five in the figure represents an SE Linux platform with a TPM.

Not all combinations of operating systems and security techniques in Figure 9 are equally interesting to analyze. In the next subsections, we place several solutions deemed interesting in the order of decreased overall security level. This ordering is an educated guess. Assigning absolute or relative security values to these combinations is a task we are not currently prepared to undertake.

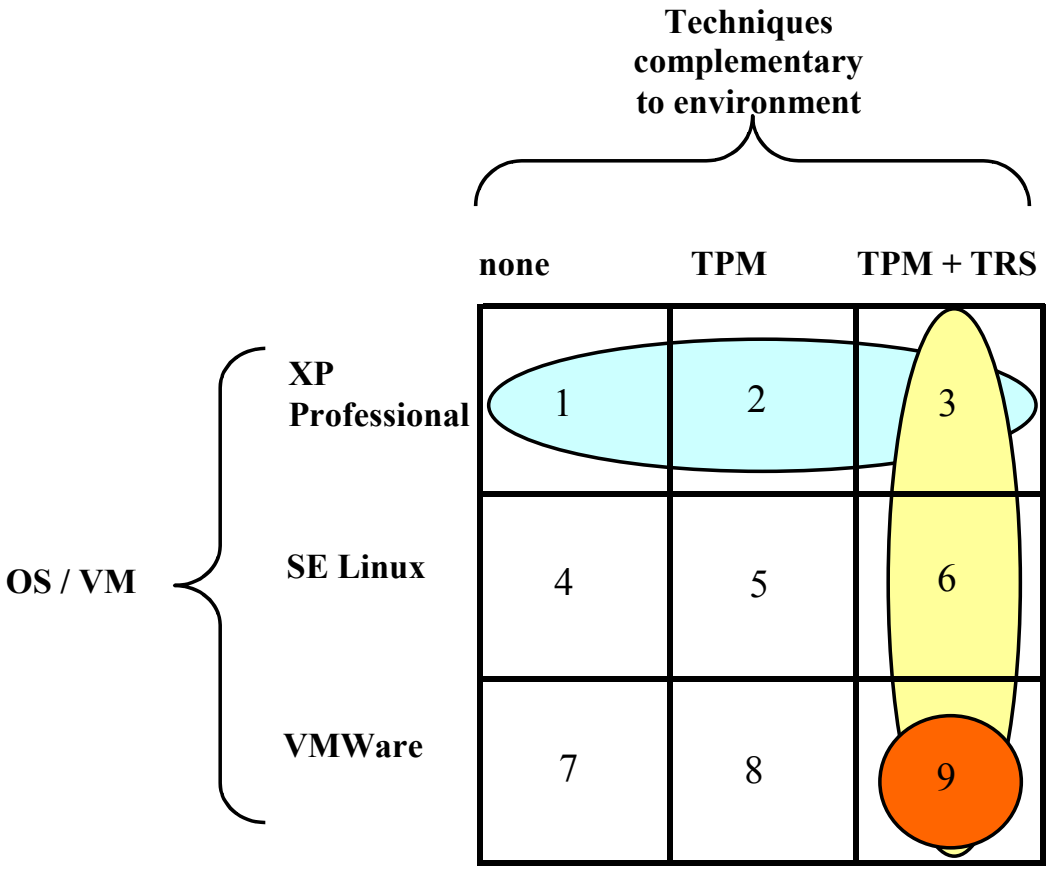


Figure 9. Operating system environment vs. security enhancement matrix

6.4.1 TRS, TPM and VMware (cell 9)

For highest possible security given the technologies available, we conceived this particular combination of technologies. The SIM software solution would be implemented using TPM. In case of a stand-alone TPM on the platform, it can assist in storing the limited amount of data and for sealing the larger bulks of data. When the TCPA compliant support is also available (BIOS, OS), the platform components and boot can be verified for integrity.

In addition to this, the TRS methods can be used. This would ensure excellent security for storage and execution when combined with TPM capabilities. One could contend that

the breaking of this combination of security to guess the SIM secret key K_i is a difficult task if the latest advances in TRS are employed. However one of the caveats in this solution is the lack of proper evaluation of TRS security methods due to the very nature of secrecy surrounding the methods and algorithms employed.

Further protection is provided when the SIM solution is implemented within an isolated execution environment making use of VMware. The host OS in this case should provide trusted boot services and maximum virtualization of hardware and system functionalities, and the guest OS can run the SIM software in an isolated partition or virtual machine. Note, that this assumes that modifications are made to host OS to provide integrity metrics. See the cost tradeoff section above for more details. Although VMware does not create strong isolation of the execution environments, it has an advantage of demonstrative compartmentalization that may prove to be beneficial for the future hardware secured implementation of SIM. In addition, design analysis is required to determine the appropriate choice of host and guest operating systems and their administrative policies. See the administrative tradeoff section above for more details.

6.4.2 TRS and TPM

This solution uses the TRS methods in conjunction with a TPM to provide secure storage and secure execution. Compared to the solution described above (Section 6.4.1) , we lose the isolated execution environment provided by VMware. We also might potentially lose the demo-ability of isolated environments by sacrificing VMware. This said it is still a strong security story along with the access control rights provided by the operating system. As is the case with the VMware story above, careful attention must be placed on appropriate OS enhancements and administration. See the cost and administrative tradeoff sections above.

6.4.2.1 TRS and TPM on SE Linux (cell 6)

Access Control and process isolation provided by SE Linux are very good starting points for our solution. If we add TPM support for secure storage of keys and data and TRS technology to secure keys during execution we have a very compelling story. One of the identified problems with this solution is the need for TPM drivers. Boot integrity could also be provided by modifying the kernel and re-architecting the BIOS. Once these things have been done – we have a very secure system. Hardware binding of software SIM could also be achieved by using TPM – if the solution needs it.

6.4.2.2 TRS and TPM on Windows XP (cell 3)

Process isolation and access control is weaker than that provided with SE Linux. We can overcome some of these shortcomings by using TRS and TPM technology. TRS can secure the keys and algorithms during execution so that even if someone could access this they couldn't get all the information needed. We can use TPM to store the keys and data. The TPM would authenticate the user/application before it can access the information in TPM. The advantage that this solution has over the one mentioned above is that we don't have to write device drivers for TPM. A disadvantage is that at this point trusted boot is not available with XP because we do not have access to its source code. Hardware binding could also be achieved by using TPM.

6.4.3 TPM and VMware (cell 8)

TPM and VMware combination without TRS is less than our highest ranked solution discussed in Section 6.4.1. The reason this solution is worth looking at is that some of the functionality offered by TRS methods, TPM, and OS overlap. For example, the combination of TPM and environmental isolation may provide sufficient security such that code TRS was designed to protect is not easily reachable by attackers. TRS methods have high implementation cost for the software SIM, because they require custom analysis of software components and the secrets to hide. Eliminating TRS, thus, provides a significant cost savings. The combination of TPM and VMware has advantages offered by TPM and separate (though not trusted) execution environment offered by VMware. The latter provides a benefit of demonstrated process isolation, rather than secure execution environment. With TCPA compliant platforms, TPM would also provide boot security, check of system integrity, better assist in storing data.

6.4.4 TPM, no VMware

Removing VMware from the solution described in Section 6.4.3 and represented by cell 8 in the matrix of Fig. 9 does not have a serious impact on the overall level of security. VMware brought an advantage of separation of execution environment. VMware was not created with security in mind. For example, communication between Host and Guest operating systems in VMware is not secured; there is no true memory isolation – VMware uses the Host OS memory manager. However, its removal has a significant impact on administrative options. For example, VMware allows users and service providers to protect data important to them by administrating their own a VMs.

6.4.4.1 TPM on SE Linux (cell 5)

Advantages of SE Linux were discussed in Section 5.2.1. In short, it offers fine-grained access rights for various objects and limits breaches in security to a bounded domain. TPM addresses the boot problem and assists to improve secure storage. To take advantage of TPM would require modifying the OS and creating the TPM driver – it's an expensive solution.

6.4.4.2 TPM on Windows XP (cell 2)

This solution is based on the assumption that Windows XP Professional is installed and used with full understanding of its security features. Improved storage security is achieved by using TPM. However, boot and run-time integrity are not available until TCPA compliant changes are made to BIOS and OS.

Windows XP (cells 1, 2, 3)

XP Professional is one solution that provides significant security if one makes diligent installation and administrative decisions. User accounts can be used to give "satisfactory" isolation between security components (SIM, ME, operator application, and communication). However, those decisions are so restrictive and their management so costly that users would not accept them. See the administrative discussion in section 6.3

7 References

* Windows® is a registered trademark of Microsoft Corporation.

---

[i] Margrave David, GSM Security and Encryption, [referred 30.9.1999] http://www.net-security.sk/telekom/phreak/radiophone/gsm/gsm-secur/gsm-secur.html

[ii] GSM interception -- http://www.dia.unisa.it/ads.dir/corso-security/www/CORS9900/a5/Netsec/netsec.html

[iii] Alex Biryukov, Adi Shamir and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. http://cryptome.org/a51-bsw.htm

[iv] Tik-110.551: Attacks against A5 --- http://www.hut.fi/~ltarkkal/netsec.pdf

[v] IBM Reserch News – Side Channel Attack
http://www.research.ibm.com/resources/news/20020507_simcard.shtml

[vi] [3GPP] 3rd Generation Partnership Project;Technical Specification Group SA WG3

[vii] 3GPP TS 33.102 V4.3.0 (2001-12)

[viii] A Guide to 3 rd Generation Security (3G TR 33.900 version 1.2.0)

[ix] 3GPP TS 33.120

[x] 3GPP TS 21.133

xi 3GPP TS 33.105

[xii] *3GPP2 R0032* Enhanced Subscriber Authentication (ESA) and Enhanced Subscriber Privacy (ESP)

[xiii] TCPA Design Philosophies and Concepts, Version 1.0, January 2001, http://www.trustedpc.org/home/pdf/designv1_0final.pdf

[xiv] P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner, J. F. Farrell, The "Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments," **Proceedings of the 21st National Information Systems Security Conference**, pp. 303-314, October 1998, also http://www.nsa.gov/selinus/inevit-abs.html.

[xv] Loscocco (he got it from the Orange Book)

[xvi] David Aucsmith, "Tamper Resistant Software: An Implementation", Information Hiding, First International Workshop, Springer LNCS 1174, May 1996, pp.317-333.

[xix] R. Meushaw and D. Simard. NetTop. Commercial Technology in High Assurance Applications. Tech Trend Notes, V. 9, Edition 4, Fall 2000.

[xx] Trusted Computing Platform Alliance (TCPA), version 1.1b.